

Black+ Dev Project 5

presented by: Natalie, Amadou & Keyara

Our Class Design: What You Need to Know

Inheritance

Sub Class

Super Class

Constructors

Loops

Catch and Throw Exceptions

Our class design was built off of inheritance concept like the previous projects and wanted the attributes and methods from the main Service Communicator class.

```
public class MyApp {  
    public static void main (String[] argv) {  
        java.util.Scanner userinput;  
  
        while (true) {  
            System.out.print("Command> ");  
            userinput = new java.util.Scanner(System.in);  
            String cmd = userinput.next();  
            System.out.println("You selected " + cmd + ".");  
        }  
    }  
}
```

We used subclasses and superclasses to extend from the Service Communicator.

```
public class Music extends ServiceCommunicator{  
  
    public static void main(String[] args) {  
        System.out.println(new Music("Bundles").get());}  
  
    Music(String songTitleOrArtistName) {  
        super("https://itunes.apple.com/search?term=" + songTitleOrArtistName  
+ "&limit=1");  
    }  
}
```

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;

class Exceptions extends ServiceCommunicator {
    public Exceptions(String errorMessage) {
        super(errorMessage);
        public ServiceCommunicator(String serviceURL) {
            try {
                URL url = new URL(serviceURL);
                URLConnection urlConnection = url.openConnection();
                conn = (HttpURLConnection)urlConnection;
            } catch(Exception ex) {
                System.err.println("**Error in constructor - Cannot create the URL or make the connection.");
                System.exit(1);
            }
        }
    }
}
```

```
public class Television extends ServiceCommunicator {  
  
    public static void main(String[] args) {  
        System.out.println(new Television("Hoarders").get());}  
  
    Television(String tvShowName) {  
        super("http://api.tvmaze.com/singlesearch/shows?q=SEARCH" +  
tvShowName);  
    }  
}
```

```
public class ZipCode extends ServiceCommunicator {  
  
    public static void main(String[] argv) {  
        System.out.println (new ZipCode("98118").get()); }  
  
    ZipCode(String zipcode) {  
        super("http://api.zippopotam.us/us/" + zipcode);  
    }  
}
```

Creating our repository in the command line.

```
C:\Users\keyar\Downloads>git config --global user.email "keyarafleece.com"

C:\Users\keyar\Downloads>git commit -m "first commit"
[master (root-commit) 2cff40d] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 3.png

C:\Users\keyar\Downloads>git branch -M main

C:\Users\keyar\Downloads>git remote add origin https://github.com/keyarafleece/Project5.git

C:\Users\keyar\Downloads>git push -u origin main
Select an authentication method for 'https://github.com/':
 1. Web browser (default)
 2. Personal access token
option (enter for default):
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 37.89 KiB | 12.63 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/keyarafleece/Project5.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

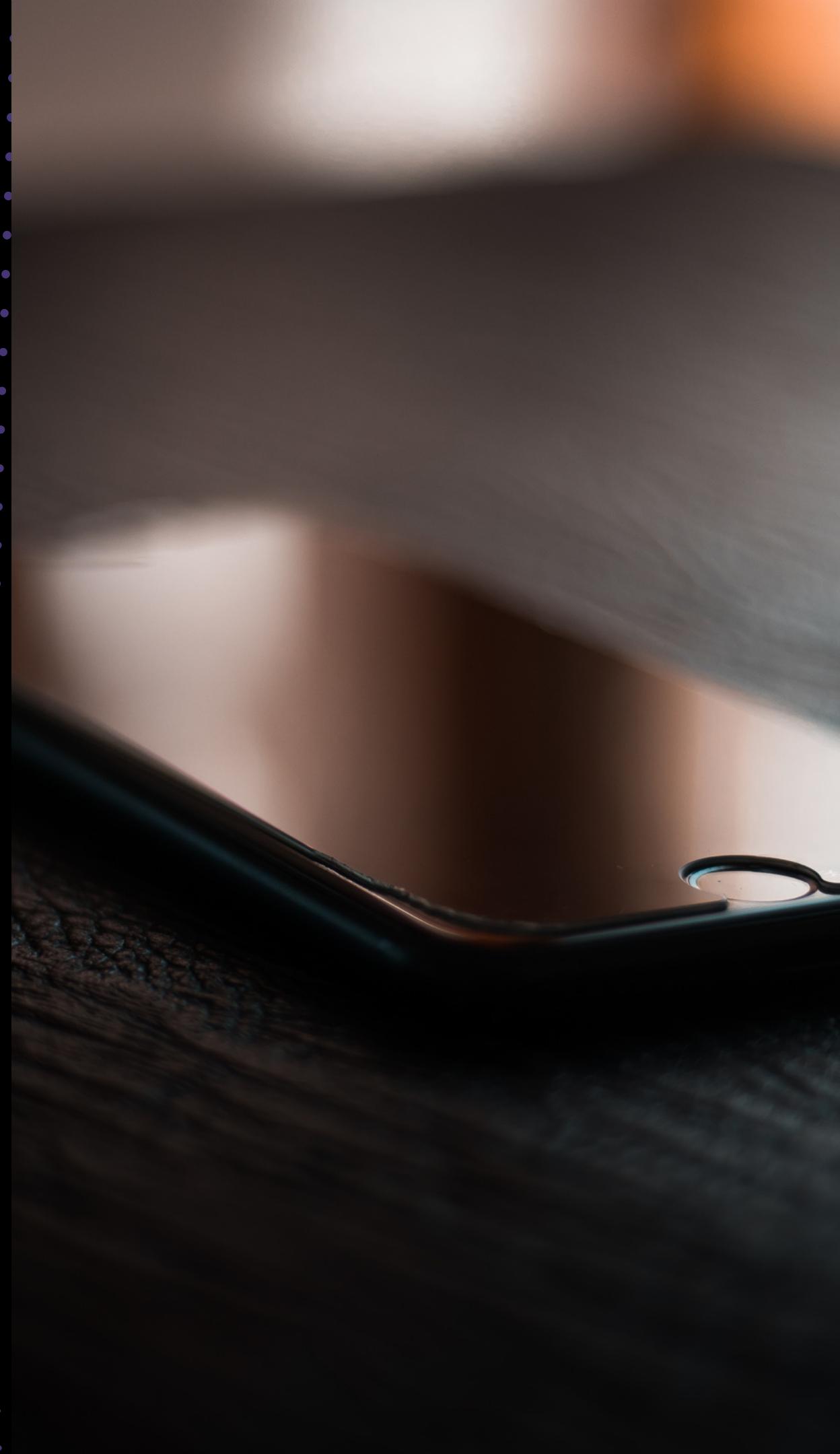
Handling Errors and Exceptions

We used the Catch and Throw Exceptions to handle errors that might arise when running our code.



Application Description

Our application retrieves information from the user to return information from an outside source.



OOP Thoughts

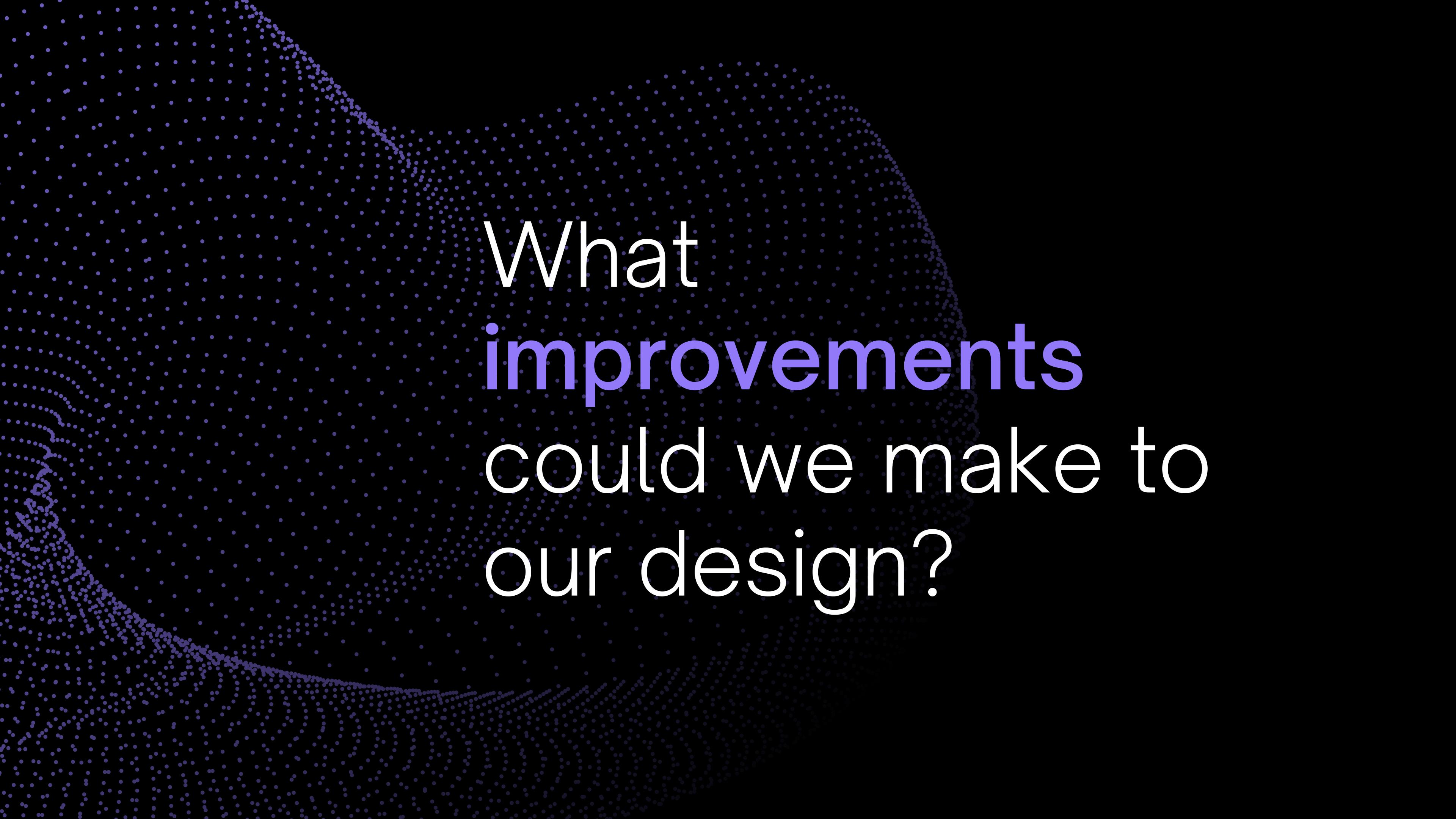
OOP is very important part of Java, simplifying software development and maintenance by different concept.

Web Services Thoughts

In order for any web service to run smoothly the Java running the design program has to be as efficient as possible.

What issues did we encounter?

GIT HUB IS A BIT OF A LEARNING CURVE BUT ONCE WE WERE ABLE TO SHARE CODE IN REAL TIME THAT MADE EVERYTHING WORK TOGETHER IN REAL TIME AND NOT MISS ON WHAT THE OTHER PERSON WAS WORKING ON.



What
improvements
could we make to
our design?

Wishlist of improvements

With more time together, we've identified these key areas of improvement for our application:

More Exceptions

We could put more exceptions in the code in order to have various responses if different errors occur

Do More With Git

We worked individually for a lot of the project but Git is a powerful collaboration tool that would have allowed us to leverage each other's talents earlier in the project.

More Websites

We could pull from more websites to see the difference in everything.

What did we **LEARN?**

How to continue to use object oriented programming to do a lot of functionality while putting GIT Hub into practice working in a group setting.

Using GitHub as a Group

GitHub is perfect for collaboration. We were able to use each other's contributions and help each other through the project.

Running Into Issues

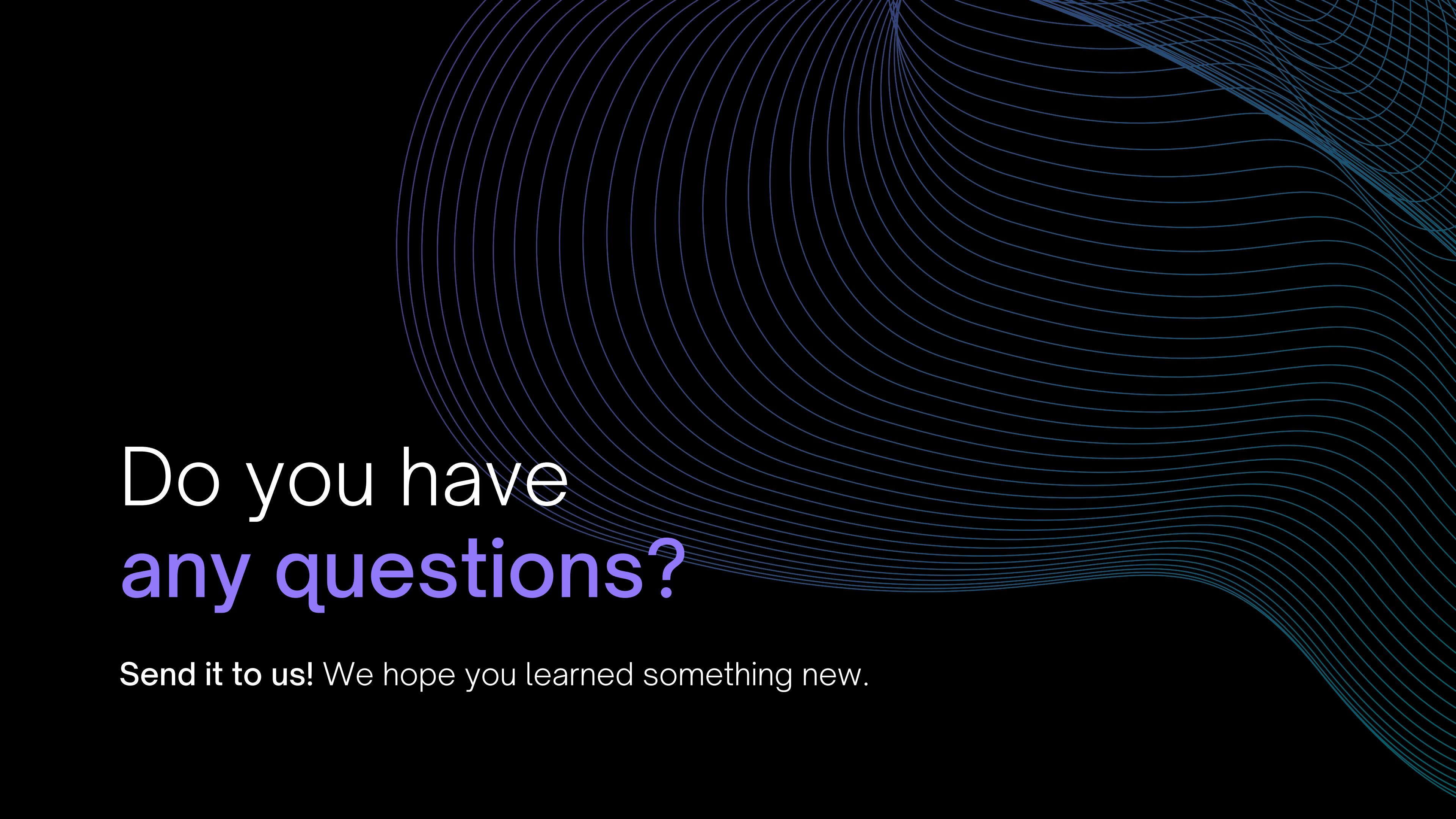
If you're learning and growing, you're going to run into issues. Congratulations, you're stretching yourself to a new level. When we ran into issues, we reached out to each other. It was very easy to be transparent on this team.

Growth Mentality

It's easy to get down on oneself when things aren't clicking immediately. Reaching out to friends and cohort mates, taking a break and coming back fresh as well as asking your tutors and instructors for a helping hand will get you through. We are here to grow and we all can do it!

“I'm not a great
programmer; I'm just a
good programmer with
great habits.”

KENT BECK



Do you have
any questions?

Send it to us! We hope you learned something new.